

MENU

SEARCH

INDEX

1/1



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 05346858

(43)Date of publication of application: 27.12.1993

(51)Int.Cl.

G06F 9/45

(21)Application number: 04179304

(71)Applicant:

SONY CORP

(22)Date of filing: 12.06.1992

(72)Inventor:

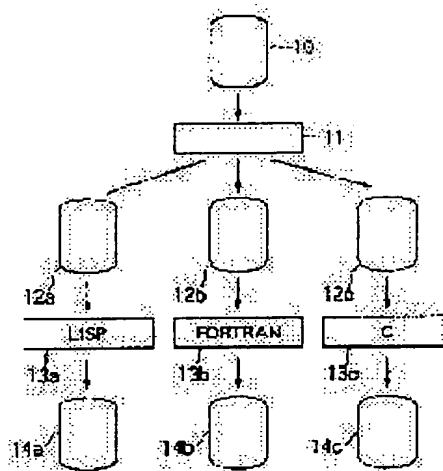
MAEKAWA HIROTOSHI
YASUDA HIROYUKI

(54) PROGRAM EDITING METHOD

(57)Abstract:

PURPOSE: To provide the program editing method which can efficiently develop a program with a single program language concerning the program conventionally developed with each of plural program languages corresponding to functions for each section in a digital computer system.

CONSTITUTION: A first compiler 11 analyzes a source program 10 containing plural functions provided with the functions of difference use for each function, and the source program is divided and converted into next-stage programs 12 (12a-12c) described in the plural program languages corresponding to the use. Further, the next-stage programs 12 are converted to machine word programs 14 (14a-14c) by second compilers 13 (13a-13c) corresponding to the respective program languages. Moreover, such a method is applied to a cross compiler environment as well, and a program used for plural computers is obtained by editing a single program on a single computer.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of
rejection]
[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

MENU

SEARCH

INDEX

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-346858

(43)公開日 平成5年(1993)12月27日

(51)Int.Cl.⁵

G 0 6 F 9/45

識別記号

庁内整理番号

F I

技術表示箇所

9292-5B

9292-5B

G 0 6 F 9/44

3 2 2 A

3 2 0 A

審査請求 未請求 請求項の数4(全 8 頁)

(21)出願番号

特願平4-179304

(22)出願日

平成4年(1992)6月12日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 前川 博俊

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 安田 弘幸

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

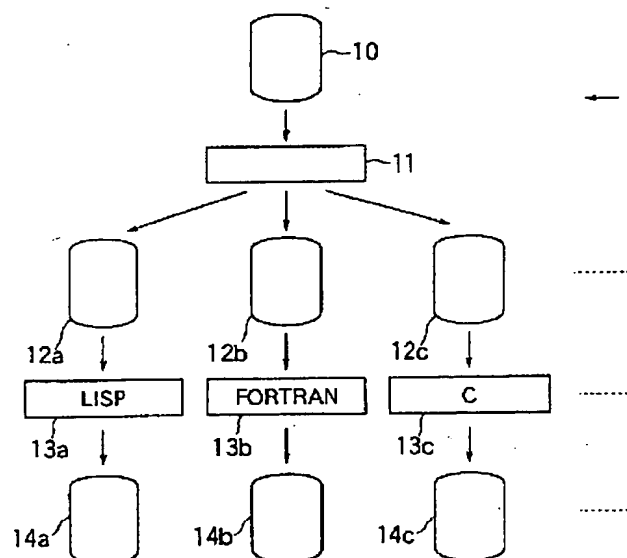
(74)代理人 弁理士 佐藤 隆久

(54)【発明の名称】 プログラム編集方法

(57)【要約】

【目的】 デジタル計算機システムにおいて、従来機能に応じて部分ごとに複数のプログラム言語で開発されていたプログラムを単一のプログラム言語で能率のよいプログラムの開発の可能なプログラム編集方法を得る。

【構成】 用途の異なった機能を持つ複数の関数を含むソースプログラム10を関数ごとに、第一のコンパイラ11で解析し、用途に合った複数のプログラム言語で記述された次段階プログラム12に分割、変換する。更に、次段階プログラム12をそれぞれのプログラム言語に対応した第二のコンパイラ13により機械語プログラム14に変換される。さらに、上記のような方法をクロスコンパイラ環境についても応用し、複数の計算機に使用するプログラムを、単一の計算機上で単一のプログラムを編集することにより得る。



【特許請求の範囲】

【請求項1】 一種類の高級プログラミング言語で記述されたソースプログラムについて、

複数の高級プログラム言語の編集機能を有し、
上記ソースプログラムの部分ごとの処理内容を分析し、
その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、
上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、
さらに、上記変換されたプログラムを機械語に変換することを特徴としたプログラム編集方法。

【請求項2】 請求項1のプログラム編集方法において、上記処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることを特徴とするプログラム編集方法。

【請求項3】 請求項2のプログラム編集方法において、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することを特徴とするプログラム編集方法。

【請求項4】 請求項3のプログラム編集方法において、一の計算機において、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、
上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、
他の計算機において、さらに、上記変換されたプログラムを機械語に変換することを特徴としたプログラム編集方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、デジタル計算機におけるコンパイラに関するものであり、特に複数のプログラム言語が同一の計算機システムで動作する場合におけるコンパイラおよびプログラム編集方法に関する。

【0002】

【従来の技術】 従来からFORTRAN等の高級プログラム言語で記述されたソースプログラムを機械語等の下位言語に変換するコンパイラといったプログラム編集方法が知られ、使用されている。以下、コンパイラを使用し、高級プログラム言語で記述されたソースプログラムを下位言語に変換する、従来のプログラム編集方法について述べる。図9は従来のコンパイラでのデータの流れを示した図である。ソースプログラム12bはFORTRANで記述されたプログラムである。コンパイラ13bはFORTRANで記述されたソースプログラムの文法チェック等を行い、機械語に翻訳する処理（コンパ

ル）を行うプログラムである。オブジェクトプログラム14bは、ソースプログラム12bをコンパイラ13bにより機械語に翻訳したプログラムである。以上の各プログラムはいずれも同一の計算機上の記憶装置上にファイルの形で存在するものである。

【0003】 計算機使用者は、エディタと呼ばれる文書作成用プログラムを使用して先ずソースプログラム12bをFORTRANで作成する。計算機使用者は、ソースプログラム12bの作成が終了した後、コンパイルを行うソースプログラム12bを指定し、コンパイラ13bを起動する。起動されたコンパイラ13bはソースプログラム12bの文法チェック、変数チェック等を行った後、ソースプログラム12bを機械語に翻訳しオブジェクトプログラム14bを作成する。

【発明が解決しようとする課題】

【0004】 従来のプログラム編集方法は以上のように構成されているため、一種類のコンパイラは一種類の高級プログラム言語しか取り扱うことができなかった。高級プログラム言語を使用した場合、ほとんどの用途のプログラムについて単一言語で対応することができる。一方、各高級プログラム言語はそれぞれ用途に向き、不向きがあり、例えば、FORTRANは数値演算に向いているが、計算機のハードウェアに密着した入出力（I/O）を取り扱うといった用途には向いていない。このため、例えば、FORTRANを使用して数値演算とI/O操作の混在したプログラムを作成すると、数値計算処理は高速だが、I/O操作処理は遅いオブジェクトプログラムが生成されるという問題が生じる。

【0005】 上記問題を解決するためには、計算機使用者がソースプログラムの仕様を分析し、数値演算部分をFORTRANで記述し、I/O操作部分を、例えばCで記述する方法がある。この方法を用いた場合、ソースプログラム仕様を分析し、各部分について異なった高級プログラム言語で記述するという高度な能力が計算機使用者に要求され、ソースプログラム開発上の大きなネックが生じるという問題点があった。本発明は、このような従来技術の問題点に鑑みてなされたものであり、単一のソースプログラムから、自動的にその部分ごとの処理に最適な高級プログラム言語を選択し、その高級プログラム言語で記述された次段階のプログラムに変換することができ、また、上記変換が複数の計算機からなる計算機ネットワークにも適用でき、しかも上記変換処理を行う際に、計算機ネットワーク全体の資源を有効利用することができるプログラム編集方法を提供することを目的とする。

【0006】

【課題を解決するための手段】 上記の問題を解決するため、本発明に係るプログラム編集方法は、一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上

記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することを特徴とする。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることを特徴とする。

【0007】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することを特徴とする。また、一の計算機において、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記変換されたプログラムを機械語に変換することを特徴とする。

【0008】

【作用】一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することにより、処理内容に応じて最適な高級プログラム言語を選択することを可能とし、また、変換後のプログラムの計算機間のポータビリティを実現することを可能としている。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることにより、標準的な各高級プログラム言語から機械語への変換用コンパイラより高度な機能および性能を持つ、上記計算機専用で作成されたコンパイラの使用を可能としている。

【0009】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することにより、上記計算機ネットワーク全体として最適なオブジェクトプログラムを得ることを可能としている。また、一の計算機において、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記変換されたプログラムを機械語に変換することにより、上記計算機ネットワーク全体としての計算機資源の有効利用が可能となる。

【0010】

【実施例】図面を参照して本発明の第一の実施例について説明する。図2は、図1(A)に示されたプロセッサエレメント(PE)の構成を示す図である。本発明は、図2に示すような記号処理用計算機(図2におけるEU)とマンマシンインターフェースおよび記憶等に使用される計算機(図2におけるRM)からなる並列計算機ネットワークのPEに適用される。エバリュエーター(EU)とリソースマネージャー(RM)はRMインターフェース(RM I/F)で接続され、EUからRMに機能データを送出する。また、EUからメインメモリインターフェース(MM I/F)を介して情報データが送出され、メインメモリ(MM)とRMに送出される。RM I/FおよびMM I/FはEU内の内部バス(IBUS)を介してALUに接続される。また、RM I/FはRM内のエバリュエーターインターフェース(EU I/F)を介してWSに接続される。EUはALU、マスカ、シフタ、ローカルメモリ、レジスタファイル、ユーザスタック、ディスパッチテーブル、インストラクションキャッシュ、システムコントローラ、コントロールメモリ、IBUS、ダイアグノスティクススタティクス、RM I/FおよびMM I/Fから構成され、RMはWSR3000、ダイアグノスティクスインターフェース、通信インターフェース、二次記憶、MM、ポインタマニピレータ、EU I/F、CRTディスプレイから構成されている。

【0011】図3は本実施例を説明するための図2の計算機ネットワークの本発明に関する部分を要約したものである。図3において、第一の計算機2は、図2のRMに、CPU20はWSに、メモリ22は、図2のMMに相当に、表示装置24は図2のCRTディスプレイに、通信インターフェース25は図2のRM I/Fに相当する。磁気記憶装置26は図2の二次記憶に相当する。なお、図2においては、バス21およびI/O23は省略されている。

【0012】図3において、CPU20は、メモリ22に蓄積されたプログラムの実行を行い、各種演算処理、および、第一の計算機2の制御を行うCPUおよびその周辺回路である。バス21は、第一の計算機2の各部分間の情報を伝達するバスである。メモリ22は、プログラムおよびデータの記憶を行うメモリである。I/O23は、表示装置24、通信インターフェース25および磁気記憶装置26の操作を行うI/O制御装置である。表示装置24は、第一の計算機2の出力データを表示するCRTディスプレイである。通信インターフェース25は他の計算機との通信を行うインターフェースである。磁気記憶装置26は、データの記憶を行う記憶装置である。

【0013】図4は本発明のプログラム編集方法のプログラム変換方法を示す図である。ソースプログラム10は、LISPで記述されたソースプログラムである。第

一のコンパイラ11は、ソースプログラム10をLISP、FORTRAN、C等の各高級プログラム言語に変換するコンパイラである。次段階プログラム12a、b、cは、ソースプログラム10の各部分部分について、第一のコンパイラ11が生成した各高級プログラム言語、LISP、FORTRANおよびCによるプログラムである。第二のコンパイラ13a、b、cは、次段階プログラム12a、b、cを機械語に変換するコンパイラである。機械語プログラム14a、b、cは、第一のコンパイラ11および第二のコンパイラ13a、b、cがソースプログラム10を変換することにより生成した機械語プログラムである。以上のソースプログラム10および各コンパイラは第一の計算機2の磁気記憶装置26上に記憶され、必要に応じてメモリ22上に読みだされる。次段階プログラム12a、b、cおよび機械語プログラム14a、b、cは、第一のコンパイラ11および第二のコンパイラ13a、b、cの動作に従い、順次生成され、メモリ22または磁気記憶装置26に記憶される。

【0014】以下、本発明のプログラム編集方法の動作について説明する。図5は本発明のプログラム編集方法のプログラム変換方法の処理のフローチャートである。ステップ01(S01)において、第一のコンパイラ11はソースプログラム10の文法チェック等を行った後、ソースプログラム10に含まれる各関数ごとに解析を行う。ソースプログラム10は図6に示すような機能を実現する関数10a～dを含んでいるものとする。ここで、関数10aは、データAを通信インターフェース25に接続される他の計算機から受信するための関数、関数10bは、データAについて、記号処理を行うための関数、関数10cは、データAについて、数値演算処理を行うための関数、関数10dは、以上の処理結果を表示装置24に表示し、磁気記憶装置26に記憶するための関数であるとする。第一のコンパイラ11はソースプログラム10の各関数10a～dに含まれる命令の種類を分類し、各関数のリスト処理用命令数、数値計算用の命令数およびI/O用の命令数をカウントする。ここで、関数10aはリスト処理用命令を3、数値計算用の命令を10、I/O用の命令を50含み、関数10bはリスト処理用命令を200、数値計算用の命令を40、I/O用の命令を0含み、関数10cはリスト処理用命令を20、数値計算用の命令を400、I/O用の命令を0含み、関数10dはリスト処理用命令を10、数値計算用の命令を10、I/O用の命令を500含んでいるという結果が生じたものとする。

【0015】ステップ02(S02)において、第一のコンパイラ11はS01の処理の結果求められた命令数の結果に基づき順次、関数10aを、I/O操作に適したプログラム言語であるCに変換するためステップ03に進む。関数10bを、リスト処理に適したプログラム

言語であるLISPに変換するためステップ04に進む。関数10cを、数値演算処理に適したプログラム言語であるFORTRANに変換するためステップ04に進む。関数10dを、I/O操作に適したプログラム言語であるCに変換するためステップ04に進む。という処理を行う。

【0016】ステップ03(S03)においては、第一のコンパイラ11がソースプログラム10より関数10bの部分を取り出し、次段階プログラム12aを得る。ここでは、ソースプログラム10がLISPで記述されているため、関数を取り出し、他の関数とのインターフェースに関する処理を行うのみで、変換は行わない。ステップ04(S04)においては、第一のコンパイラ11がソースプログラム10より関数10cの部分を取り出し、FORTRANへの変換を行い、第二の次段階プログラム12bを得る。ステップ05(S05)においては、第一のコンパイラ11がソースプログラム10の関数10aおよび関数10dの部分を取り出し、Cへの変換を行い、次段階プログラム12cを得る。ステップ05(S05)において、全関数について処理を終わったかを判断する。

【0017】以上の動作によって得られた次段階プログラム12a～cは、それぞれLISP用コンパイラ13a、FORTRAN用コンパイラ13b、C用コンパイラ13cにより、機械語プログラム14a～cに変換される。以上により、ソースプログラム10の各関数10a～dについて最適化された、第一の計算機2用の次段階プログラム12a、b、cおよび機械語プログラムa、b、cが得られる。ここで、本実施例で使用される高級プログラム言語、LISP、FORTRANおよびCの仕様は、情報処理の分野の標準的な規格に完全準拠したものである。必要な場合、他の計算機に本実施例のプログラム編集方法で生成された次段階プログラムを容易に移植可能である。

【0018】以下、第二の実施例について述べる。図7は本実施例が適用される、図2の計算機ネットワークの要点を抜き出した図である。第二の計算機3は、図2のEUに相当する。第二の計算機3は記号処理を高速に行うための専用計算機として使用される。第二の計算機3の構成は説明の簡略化のために、第一の計算機2と同様のものであると仮定する。RM I/F30およびMM I/F31は第一の計算機2と第二の計算機3を接続し、各種情報を送受信するためのインターフェースである。第一の計算機2と第二の計算機3は、このいずれかを介して必要な情報の送受信を行う。

【0019】第一の計算機2および第二の計算機3は、それぞれ第一の実施例に示した本発明に係るプログラム編集方法を実施するための計算機環境を持っているものとする。ただし、第二の計算機3の第一のコンパイラ11、LISP用コンパイラ13aおよびFORTRAN

用コンパイラ13bは、第二の計算機3用に特化し、強化された独特の命令に対応したものであるとする。上記両計算機において、それぞれ第一の計算機2用に開発されたソースプログラム10および第二の計算機3用に開発されたソースプログラム10を第一の実施例と同様の方法で機械語プログラム14a、b、cに変換する。以上により、計算機の種類に応じて、最も最適化された機械語プログラム14a、b、cを得ることができる。

【0020】以下、第三の実施例について述べる。本実施例はいわゆるクロスコンパイル環境を提供するものである。図8は本実施例のプログラム編集方法のプログラム変換方法を示す図である。本実施例のプログラム編集方法の動作計算機環境は、図7の第一の計算機2に設けられる。ソースプログラム10は、LISPで記述され、第一の計算機2および第二の計算機3上で動作する関数が混在し、その関数ごとに、一定のフォーマットで前記いずれの計算機で動作するかを設定してあるものとする。第一のコンパイラ11aは、ソースプログラム10の上記設定に基づき、ソースプログラム10の各部分を第一の計算機2用と第二の計算機3用に分割する。ソースプログラム10a、bは、それぞれ第一の計算機2用、第二の計算機3用に分割されたソースプログラム10である。第一のコンパイラ11a、bは、それぞれ第一の計算機2用、第二の計算機3用に特化した第一のコンパイラ11である。第一のコンパイラ11bは第二の計算機3用に強化した各プログラム言語の命令に対応しているものとする。次段階プログラム12a、b、cは第一の計算機2用の次段階プログラムである。次段階プログラム12d、e、fは第二の計算機3用の次段階プログラムである。第二のコンパイラ13a、b、cは、第一の計算機2用のLISP用、FORTRAN用、C用コンパイラである。第二のコンパイラ13d、e、fは、第二の計算機3用のLISP用、FORTRAN用、C用コンパイラである。これらは、第二の計算機3用に強化された命令に対応しているものとする。

【0021】次に動作を説明する。ソースプログラム10は、第一のコンパイラ11bにより、第一の計算機2用および第二の計算機3用の各部分に分割される。この際、第一のコンパイラ11bはソースプログラム10作成時に挿入された上記区別に従い、関数ごとに分割するものとする。この結果、ソースプログラム10a、bが生成される。このソースプログラム10a、bについて、第一の実施例と同様の手順により最終的な機械語プログラム14a～fが生成される。このうち機械語プログラム14d～fは第二の計算機3用のものである。機械語プログラム14d～fは第二の計算機3上に移され、動作する。本実施例では、予め上記区別をソースプログラム10に挿入したが、第一のコンパイラ11aにおいて、自動的に区別するように構成してもよい。また、最初からソースプログラム10a、bを別々に開発

してもよい。

【0022】以下、第四の実施例を説明する。図8において、第一の計算機2には第二のコンパイラ13d～fを除いたプログラム編集方法計算機環境を、第二の計算機3には第二のコンパイラ13d～fのみを用意する。第一の計算機2においては、第一の計算機2用の機械語プログラム14a、b、cを第一の実施例と同様に作成する。また、第三の実施例と同様の方法で第二の計算機3用の次段階プログラム12d～fを作成し、第二の計算機3に移す。第二の計算機3上では、次段階プログラム12d～fを第二のコンパイラ13d～fにかけ、最終的な機械語プログラム14d～fを作成する。

【0023】本発明のプログラム編集方法は、上記実施例に限定されず、他に種々の構成をとることができる。また、上記した計算機環境は例示である。特に、計算機の数、種類、第一のコンパイラ11の生成する次段階プログラム12に使用されるプログラム言語、ソースプログラム10のプログラム言語はここで述べたものに限らない。

20 【0024】

【発明の効果】以上述べたように本発明のプログラム編集方法によれば、一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することにより、処理内容に応じて最適な高級プログラム言語を選択することを可能とし、また、変換後のプログラムの計算機間のポータビリティを実現したプログラム編集方法を提供する。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることにより、標準的な各高級プログラム言語から機械語への変換用コンパイラより高度な機能および性能を持つ、上記計算機専用で作成されたコンパイラの使うことができるプログラム編集方法を提供する。

30 【0025】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することにより、上記計算機ネットワーク全体として最適なオブジェクトプログラムを得ることができるプログラム編集方法を提供する。また、一の計算機において、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記
50 変換されたプログラムを機械語に変換することにより、

上記計算機ネットワーク全体としての計算機環境の有効利用を図ることができるプログラム編集方法を提供する。

【図面の簡単な説明】

【図1】本発明が適用される計算機ネットワークの接続形態を示す図である。

【図2】本発明が適用される計算機ネットワークの構成図である。

【図3】本発明が適用される計算機の構成図である。

【図4】本発明のプログラム編集方法のプログラム変換方法を示す図である。

【図5】本発明のプログラム編集方法のプログラム変換方法の処理のフローチャートである。

【図6】ソースプログラムの構成を示す図である。

【図7】本発明の第三実施例の計算機ネットワークの構成を示す図である。

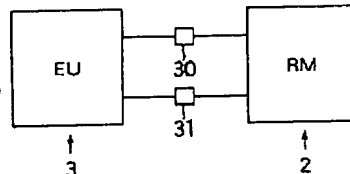
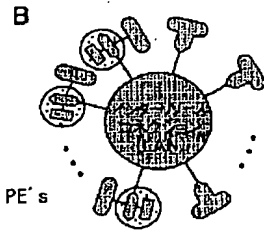
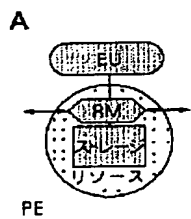
【図8】本発明の第三の実施例のプログラム編集方法のプログラム変換方法を示す図である。

【図9】従来のプログラム編集方法を示す図である。

【符号の説明】

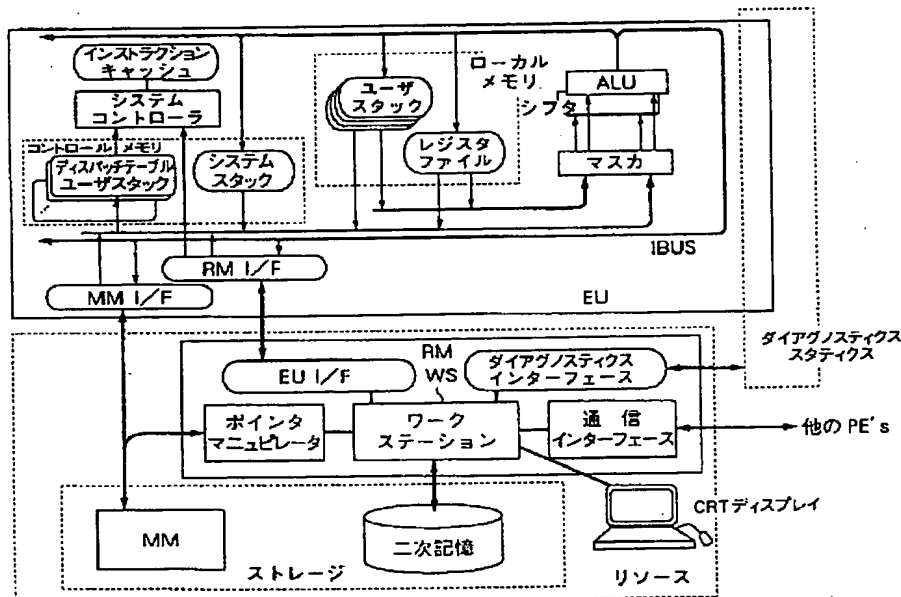
- 1・・・本発明のプログラム編集方法
- 2・・・本発明が適用される第一の計算機
- 3・・・本発明が適用される第二の計算機
- 10・・・ソースプログラム
- 11・・・第一のコンパイラ
- 12・・・次段階の高級プログラム言語によるプログラム
- 13・・・第二のコンパイラ
- 14・・・最終的な機械語プログラム
- 20・・・CPU
- 21・・・バス
- 22・・・メモリ
- 23・・・I/O
- 24・・・表示装置
- 25・・・通信インターフェース
- 26・・・磁気記憶装置
- 30・・・RM I/F
- 31・・・MM I/F

【図1】

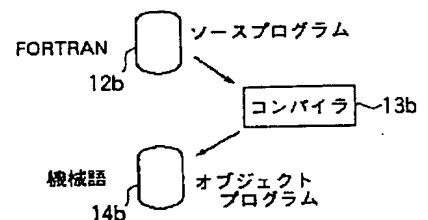


【図7】

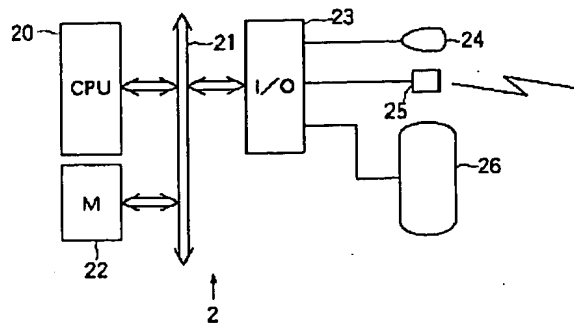
【図2】



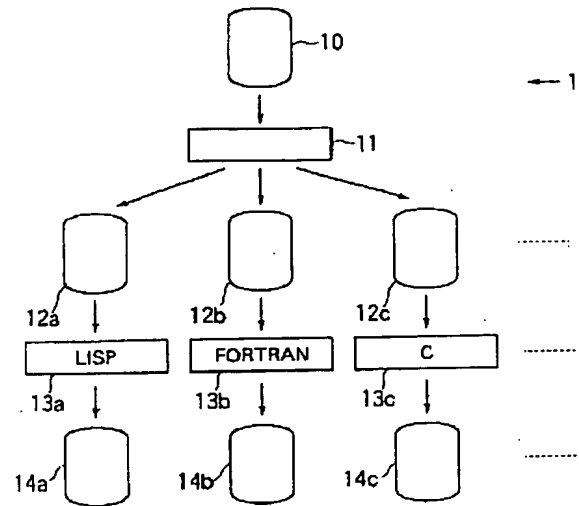
【図9】



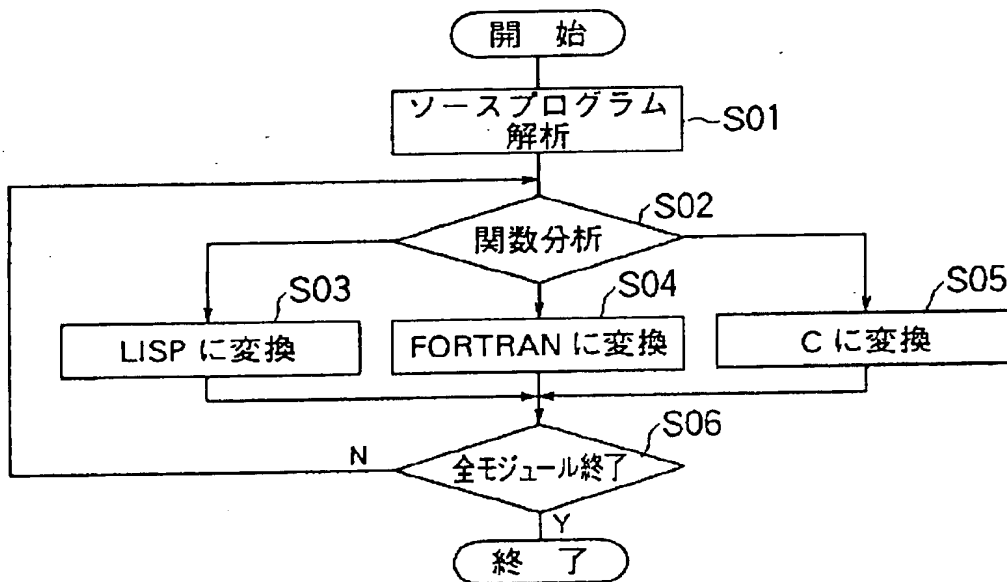
【図3】



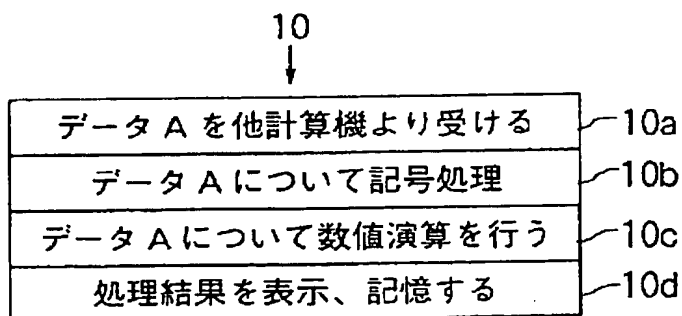
【図4】



【図5】



【図6】



【図8】

